Indian Institute of Technology, Gandhinagar

Ball & Beam Part-02

ES245 Control Systems

Team Vadapav - Group 04

Authors	*Roll Number:*
Deepak soni	22110068
Shaurykumar Patel	22110241
Piyush singh	22110253
Sridhar Singh	22110257

November 27, 2024

Contents

1 Introduction **2**

2 Setup Overview **3**
2.1 List of Materials 3
2.2 Hardware Overview 3

3 Essential Background **4**

4 Root Locus and Bode Plot of Ball-Beam System **6**
4.1 **Task 6.1** 6
4.2 **Task 6.2** 7

5 Controller Design using Root Locus and Bode Plot Approaches **9**
5.1 **Task 7.1** 9
5.2 **Task 7.2** 11
5.3 **Task 7.3** 12
5.3.1 Root Locus : 12
5.3.2 Bode Plot : 13

6 Final Prototype **14**

7 Conclusion **15**

8 Acknowledgement **15**

1 Introduction

The ball and beam system is a classical control problem that demonstrates the concepts of feedback control and system dynamics. In this system, a ball is placed on a beam and is allowed to roll freely under the influence of gravity along the beams length. The beam's angle can be controlled by a servo motor, which adjusts the inclination of the beam through a lever arm mechanism. The primary objective of this project is to design a controller that can stabilize the ball at a desired position on the beam by manipulating the servo motor's angle.

This project challenges students to model the dynamics of the ball and beam system, analyze its stability, and implement different control strategies. Given the non-linear nature of the system, the behavior of the ball is influenced by factors such as its mass, radius, moment of inertia, and the geometry of the beam. The system's complexity requires a step-by-step approach to linearization, transfer function analysis, and the design of an effective feedback controller.

The ultimate goal is to use a proportional-integral-derivative (PID) controller to manipulate the balls position, minimizing overshoot, reducing settling time, and ensuring system stability. By simulating the system in MATLAB and Simulink, and later constructing a physical prototype, students will explore real-world challenges in control systems, such as tuning PID gains and achieving the desired performance criteria.

This project is not only an exploration of control theory but also a practical exercise in system design, modeling, and implementation. The final deliverable will include a technical report, CAD drawings, programming scripts, and a physical system capable of controlling the balls position on the beam using a PID controller.

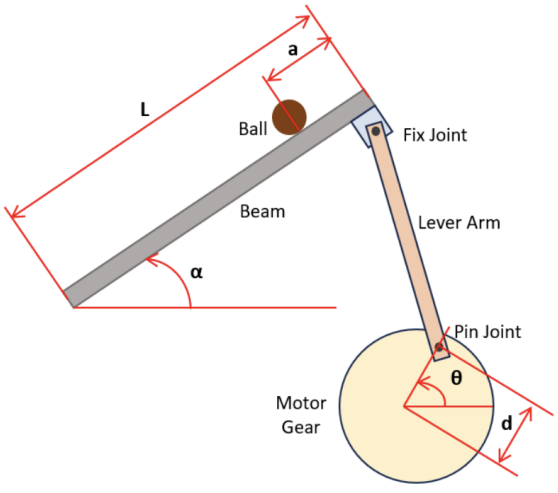

Figure 1: Ball and Beam schematic

2 Setup Overview

2.1 List of Materials

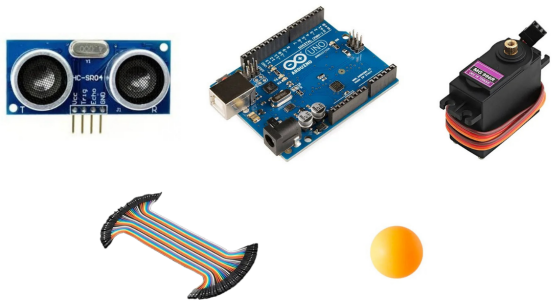

Figure 2: Materials

2.2 Hardware Overview

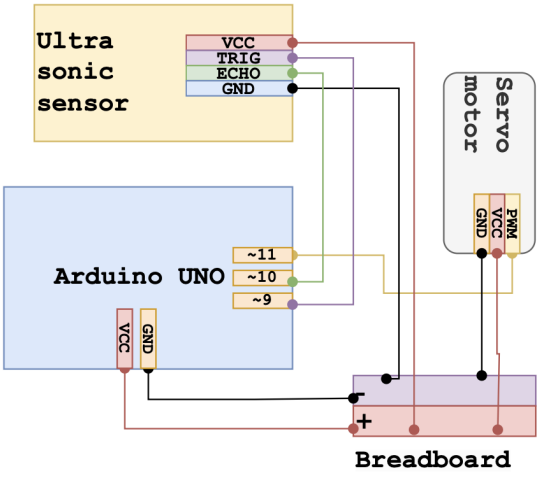

Figure 3: Circuit diagram

This was the electrical/hardware setup used in our physical ball and beam model.

- Ultrasonic sensor were used for measuring distance and based on changing the level of beam.
- Servo motor were used for tilting beam so that ball can be balanced at a certain point.
- Arduino UNO is used for fetching code and giving the input to the servo motor based on ultrasonic sensor

3 Essential Background

As shown in the figure below, a ball is placed on a beam, where it rolls under the influence of gravity. The ball has one degree of freedom along the length of the beam. The beam's angle is adjusted using a lever arm connected to the beam at one end and to a servo motor gear at the other. As the servo gear rotates by an angle θ about its axis, the lever arm angle changes by α from the horizontal.
Where :

- m is the mass of ball (in Kg)
- R is the radius of ball (in m)
- α is the angle of beam (in rad)
- J is the moment of inertia of ball (in $Kg{-}m^2$)
- J_m combined moment of inertia of ball and beam (in $Kg - m^2$)

Assumptions : The ball remains in contact with the beam at all times. The ball performs pure rolling motion (rolling without slipping).

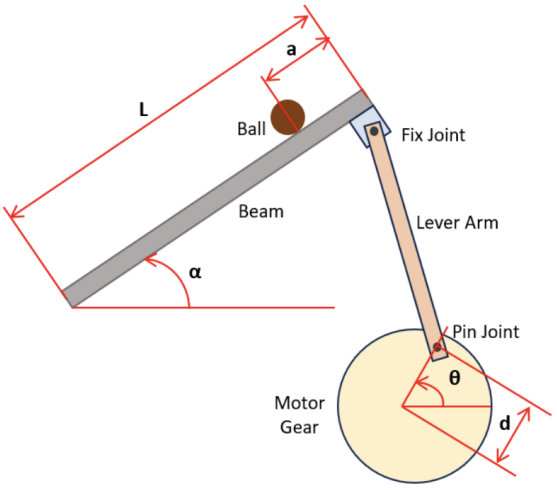

Figure 4: Ball and Beam Schematic

$$\text{Total Potential Energy(V)} = mgR\sin\alpha + \frac{1}{2}MgL\sin\alpha$$

$$\text{Total Kinetic Energy(T)} = \frac{1}{2}m(\dot{r}^2 + r^2\dot{\alpha}) + \frac{1}{2}J(\frac{\dot{r}}{R})^2 + \frac{1}{2}J_m(\dot{\alpha})^2$$

$$\text{Lagrangian (L)} = \text{T-V}$$

$$L = \frac{1}{2}m(\dot{r}^2 + r^2\dot{\alpha}) + \frac{1}{2}J(\frac{\dot{r}}{R})^2 + \frac{1}{2}J_m(\dot{\alpha})^2 - mgR\sin\alpha + \frac{1}{2}MgL\sin\alpha$$

$$\text{Using } \frac{d}{dt}(\frac{\partial L}{\partial \dot{r}}) - \frac{\partial L}{\partial r} = 0$$

$$(\frac{J}{R^2} + m)\ddot{r} + mg\sin\alpha - m\dot{r}\alpha^2 = 0$$

Linearization of this equation about the beam angle, $\alpha = 0$, gives us the following linear approximation of the system:

$$\left(\frac{J}{R^2} + m\right)\ddot{r} = -mg\alpha$$

Substituting $\alpha = \frac{d}{L}\theta$ in above equation, we get

$$\left(\frac{J}{R^2} + m\right)R(s)s^2 = -mg\frac{d}{L}\Theta(s)$$

Taking the Laplace transform of the equation above, the following equation is found:

$$\left(\frac{J}{R^2}+m\right)R(s)s^2 = -mg\frac{d}{L}\Theta(s)$$

transfer function from the gear angle ($\Theta(s)$) to the ball position ($R(s)$).

$$P(s) = \frac{R(s)}{\Theta(s)} = -\frac{mgd}{L\left(\frac{J}{R^2}+m\right)}\frac{1}{s^2} \qquad [\frac{m}{rad}]$$

The system dynamics equations are as follows:

$$(\frac{J}{R^2}+m)\ddot{r} + mg\sin\alpha - m\dot{r}\alpha^2 = 0$$

The system's transfer function between motor gear angle θ and ball position (a) is as follows:

$$P(S) = \frac{R(S)}{C(S)} = -\frac{mgd}{L(\frac{J}{R^2}+m)}\frac{1}{S^2}$$

Linearizing the system equations and writing in the state space form.

$$\left(\frac{J}{R^2}+m\right)\ddot{r} = -mg\frac{d}{L}\theta$$

The linearized system equations can also be represented in state-space form.This can be done by selecting the ball's position (r) and velocity ($\dot{r}$) as the state variable and the gear angle (θ) as the input. The state-space representation is shown below:

$$\begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} r \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{mgd}{L\left(\frac{J}{R^2}+m\right)} \end{bmatrix}\theta$$

4 Root Locus and Bode Plot of Ball-Beam System

4.1 Task 6.1

Perform the following analysis using MATLAB.

- Plot root locus and comment on the systems stability. From the below plot we can say that
  1. The poles are purely imaginary (located on the imaginary axis at the origin).
  2. The system is marginally stable because the poles are not in the left half of the s-plane. This indicates that any disturbance will neither decay nor grow but will persist indefinitely.

```
m = 0.0027;
R = 0.02;
g = -9.8;
L = 0.342;
d = 0.064123;
J = 4.32e-7;
s = tf ('s') ;
r = -m * g * d / L /( J / R ^2+ m );
P_ball = -m * g * d / L /( J / R ^2+ m ) / s ^2;
rlocus(P_ball)
title('Root Locus of Ball-Beam System', 'FontSize',
   14, 'FontWeight', 'bold');
```

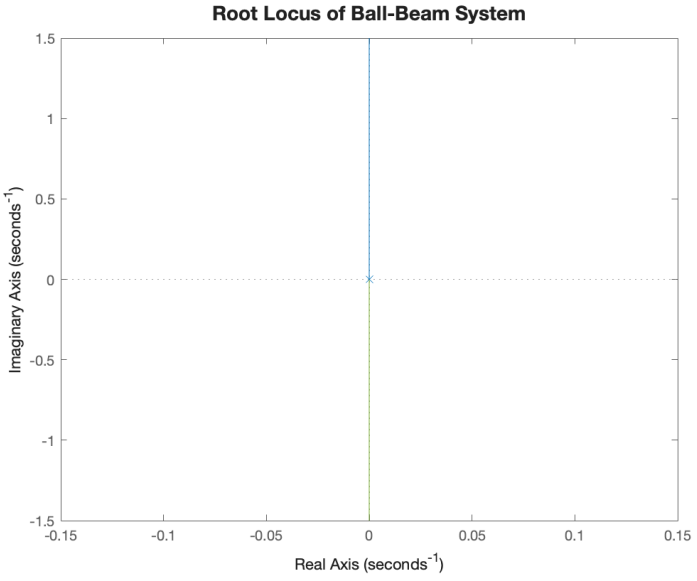

Figure 5: Root Locus of the Ball-Beam system

- Plot the bode plot and report the phase margin and gain margin of the system. Using the below Bode plot, We can say that **Phase Margin:** 0° and **Gain Margin:** ∞

```
m = 0.0027;
R = 0.02;
g = -9.8;
L = 0.342;
d = 0.064123;
J = 4.32e-7;
s = tf ('s') ;
P_ball = -m * g * d / L /( J / R ^2+ m ) / s ^2;
bode(P_ball)
title('Bode Plot of Ball-Beam System', 'FontSize',
   14, 'FontWeight', 'bold'
```

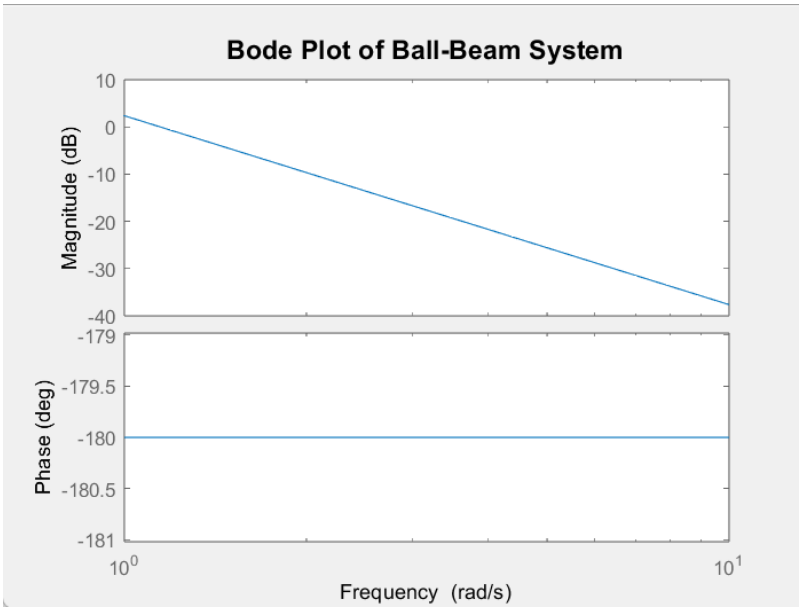

Figure 6: Bode Plot of the Ball-Beam system

4.2 Task 6.2

Perform the following analysis using MATLAB.

- **Root Locus:**Plot the design criteria using MATLAB (Hint: use sgrid command) and recommend the appropriate compensator to achieve the desired performance.

```
m = 0.0027;
R = 0.02;
g = -9.8;
L = 0.342;
```

```
d = 0.064123;
J = 4.32e-7;
s = tf ('s') ;
r = -m * g * d / L /( J / R ^2+ m );
P_ball = -m * g * d / L /( J / R ^2+ m ) / s ^2;
rlocus(P_ball)
title('Root Locus of Ball-Beam System', 'FontSize',
   14, 'FontWeight', 'bold');
sgrid(0.70,1.9)
axis([-5 5 -2 2])
```

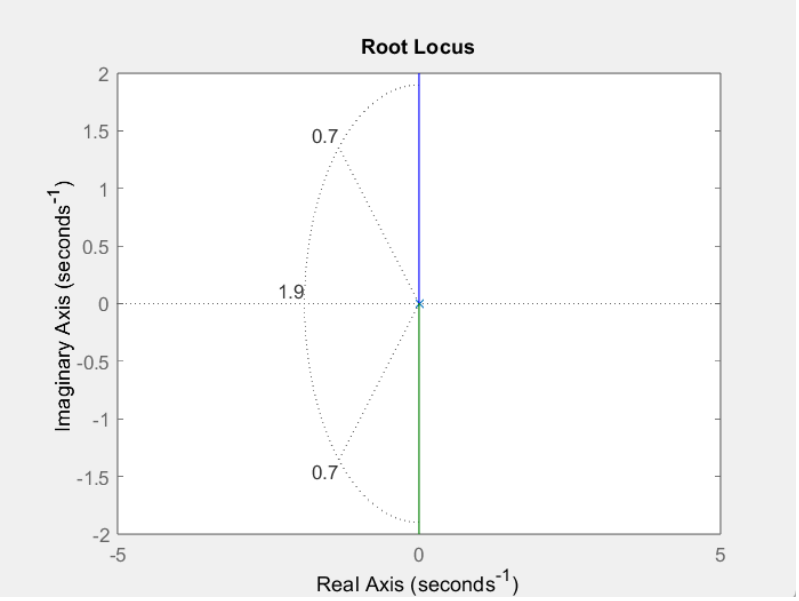

Figure 7: Root Locus with design criteria of Overshoot less than 5% and settling time less than 3 sec.

The space between the two dotted diagonal lines represents the region where the systems overshoot stays below 5%. The area outside the curved line shows where the settling time is less than 3 seconds. However, the current root locus does not meet these design requirements. To fix this and make the system stable by shifting the root locus further into the left-hand side of the plane, we will add a lead compensator.

- **Bode plot:** Analyse the plot to recommend the compensator to meet the design criteria. From the plot, we can see that the phase margin is zero. Phase

margin measures how much the system's phase can change before it becomes unstable. Since the phase margin is zero, the system is already unstable. To fix this, we need to increase the phase margin, and we can do this by using a lead compensator controller.

5 Controller Design using Root Locus and Bode Plot Approaches

5.1 Task 7.1

Based on the previous recommendations, design a first-order lead/lag compensator to meet the design criteria:

- Plot the root locus of the system with the compensator.

```
m = 0.0027;
R = 0.02;
g = -9.8;
L = 0.342;
d = 0.064123;
J = 4.32e-7;
s = tf ('s') ;
P_ball = -m * g * d / L /( J / R ^2+ m ) / s ^2;
zo = 0.005;
po = 4.79;
C=tf([1 zo],[1 po]);
rlocus(C*P_ball)
sgrid(0.70, 1.9)
k=6.433;
sys_cl=feedback(k*C*P_ball,1);
figure
info = stepinfo(sys_cl);
fprintf('Maximum overshoot time : %.2f %%\n',info.
   Overshoot);
fprintf('Settling time : %.4f seconds\n',info.
   SettlingTime);
```

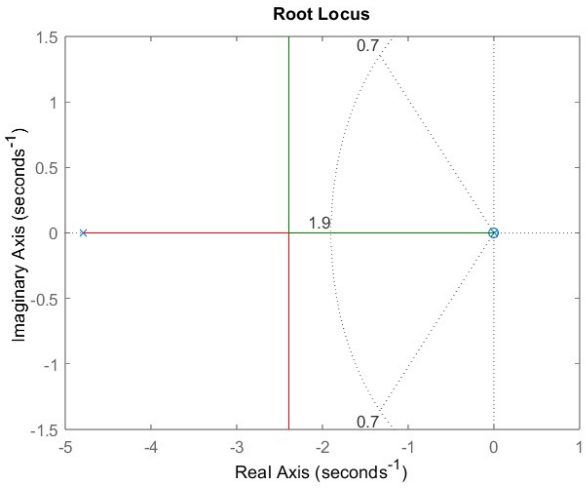

Figure 8: Root Locus with the lead Compensator

- Plot the bode plot of the system with the compensator.

```
m = 0.0027;
R = 0.02;
g = -9.8;
L = 0.342;
d = 0.064123;
J = 4.32e-7;
s = tf('s');
P_ball = -m * g * d / L / (J / R^2 + m) / s^2;
% Phase-lead compensator design

phi = 85; % Updated desired phase margin (degrees)
wbw = 1.9; % Bandwidth frequency (rad/s)
pmr = phi * pi / 180;

a = (1 - sin(pmr)) / (1 + sin(pmr)); % Compute 'a'
T = sqrt(a) / wbw; % Compute 'T'
aT = 1 / (wbw * sqrt(a)); % Compute 'aT'
K = 1.15; % Gain

C = K * (1 + aT * s) / (1 + T * s); % Compensator
   transfer function

% Plot root locus with compensator
figure;
rlocus(C * P_ball);
sgrid(0.70, 1.9); % Add damping ratio and natural
   frequency grid
title('Root Locus with Compensator');

% Closed-loop system
sys_cl = feedback(C * P_ball, 1);

% Step response information
info = stepinfo(sys_cl);
fprintf('Maximum overshoot: %.2f %%\n', info.
   Overshoot);
fprintf('Settling time: %.4f seconds\n', info.
   SettlingTime);

% Bode plot of the open-loop system
figure;
bode(C * P_ball);
grid on;
title('Bode Plot of Open-Loop System');
```

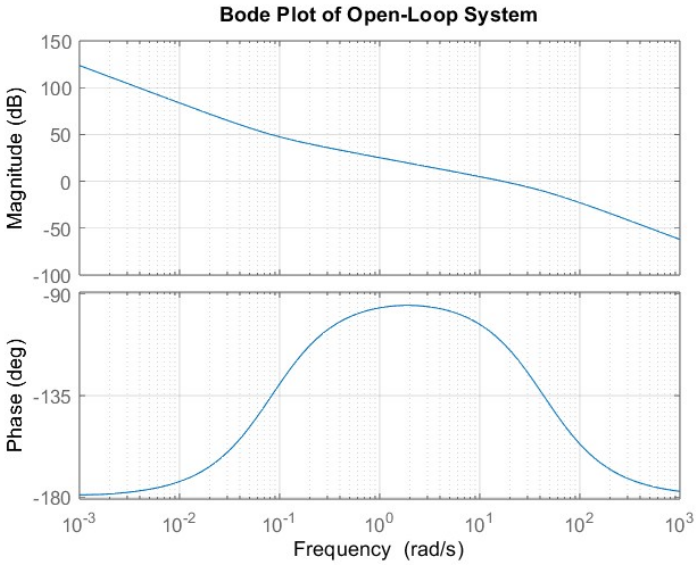

Figure 9: Bode Plot with the lead Compensator

5.2 Task 7.2

Comment on the pole-zero placement strategy used in both approaches and explain how the compensator pole-zero placement affects the root locus and bode plot.

In **Task 7.1.1**, the compensator design places the zero close to the origin and the pole farther away. This placement significantly impacts the **root locus** by pulling the branches towards the left-half s-plane, improving system stability and damping. This ensures a better transient response by reducing oscillations and achieving the desired performance. The simple pole-zero placement results in an effective and straightforward design that meets the transient performance requirements.

In **Task 7.1.2**, the compensator is optimized for specific **Maximum Overshoot** and **Settling Time** requirements. The zero near the origin enhances the mid-frequency gain and provides a phase lead, which increases the phase margin. The pole, positioned farther to the left, ensures the high-frequency gain rolls off, maintaining stability and robustness. This configuration has a clear impact on the **Bode plot**, with a noticeable phase boost in the mid-frequency range, improving stability and ensuring the desired phase margin and bandwidth are achieved. The compensator design allows for precise tuning of the systems performance metrics, including phase margin, bandwidth, and transient response.

Overall, both designs leverage strategic pole-zero placement to enhance system stability and meet specific performance objectives, with Task 7.1.1 focusing on transient response improvement via root locus and Task 7.1.2 targeting phase margin optimization via Bode plot adjustments.

5.3 Task 7.3

Plot the systems closed-loop response (with the compensator) for a step input. Verify that the system satisfies the design criteria using both the root locus and bode plot approaches.

5.3.1 Root Locus :

Maximum Overshoot : 1.31% and Settling time : 1.3469 sec

```
m = 0.0027;
R = 0.02;
g = -9.8;
L = 0.342;
d = 0.064123;
J = 4.32e-7;
s = tf ('s') ;
P_ball = -m * g * d / L /( J / R ^2+ m ) / s ^2;
zo = 0.005;
po = 4.79;
C=tf([1 zo],[1 po]);
rlocus(C*P_ball)
sgrid(0.70, 1.9)
k=6.433;
sys_cl=feedback(k*C*P_ball,1);
t=0:0.01:5;
figure
step(0.25*sys_cl,t)
info = stepinfo(sys_cl);
fprintf('Maximum overshoot time : %.2f %%\n',info.
   Overshoot);
fprintf('Settling time : %.4f seconds\n',info.
   SettlingTime);
```

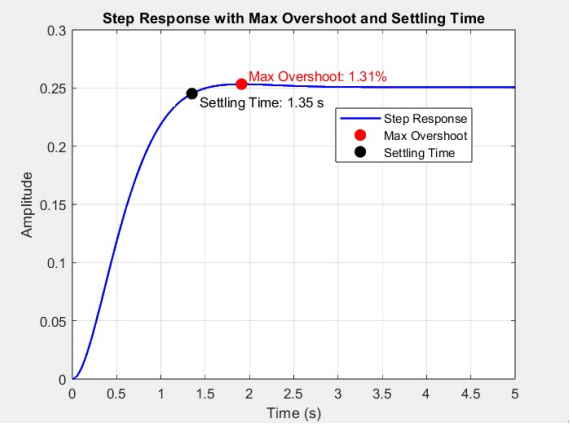

Figure 10: Step Response for a step input with Compensator designed using Root Locus.

5.3.2 Bode Plot :

Maximum Overshoot : 2.62% and Settling time : 0.2116 sec

```
% Constants for the system
m = 0.0027; % mass of the ball
R = 0.02;  % radius of the ball
g = -9.8;  % gravitational acceleration
L = 0.342; % length of the ramp
d = 0.064123; % distance from pivot
J = 4.32e-7; % moment of inertia

s = tf('s');
P_ball = -m * g * d / L / (J / R^2 + m) / s^2; % Plant
   transfer function

% Phase-lead compensator design
phi = 85; % Updated desired phase margin (degrees)
wbw = 1.9; % Bandwidth frequency (rad/s)
pmr = phi * pi / 180; % Convert phase margin to radians

a = (1 - sin(pmr)) / (1 + sin(pmr)); % Compute 'a'
T = sqrt(a) / wbw; % Compute 'T'
aT = 1 / (wbw * sqrt(a)); % Compute 'aT'
K = 1.15; % Gain

C = K * (1 + aT * s) / (1 + T * s); % Compensator
   transfer function

% Plot root locus with compensator
figure;
rlocus(C * P_ball);
sgrid(0.70, 1.9); % Add damping ratio and natural
   frequency grid
title('Root Locus with Compensator');

% Closed-loop system
sys_cl = feedback(C * P_ball, 1);

% Step response
figure;
t = 0:0.01:5;
step(0.25 * sys_cl, t);
title('Step Response with Compensator');
grid on;

% Step response information
info = stepinfo(sys_cl);
```

```
fprintf('Maximum overshoot: %.2f %%\n', info.Overshoot);
fprintf('Settling time: %.4f seconds\n', info.
   SettlingTime);

% Bode plot of the open-loop system
figure;
bode(C * P_ball);
grid on;
title('Bode Plot of Open-Loop System');
```

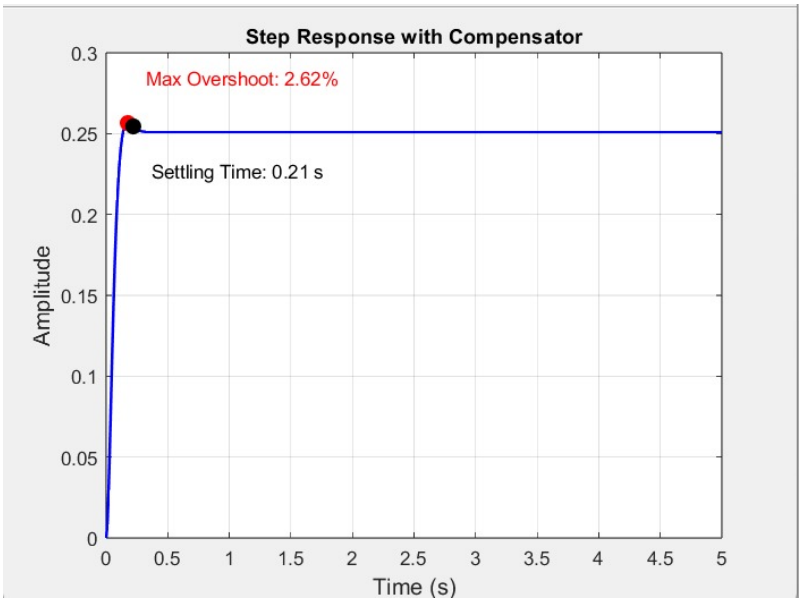

Figure 11: Step Response for a step input with Compensator designed using Bode Plot.

6 Final Prototype

Figure 12: CAD Model

Watch Video: PID Control

7 Conclusion

The Ball and Beam system is a fascinating control problem that demonstrates fundamental concepts in control theory, particularly when balancing a ball on a beam. The challenge lies in designing a control system that can stabilize the inherently unstable system. Among the various control strategies, the PID controller plays a crucial role, with the proportional (K_p), integral (K_i), and derivative (K_d) gains being critical parameters that must be carefully tuned. Tuning these parameters correctly is essential for ensuring system stability, fast response, and minimizing overshoot and steady-state error. The proportional term influences the system's reaction to current errors, the integral term helps eliminate steady-state errors, and the derivative term predicts future errors based on the rate of change. An improper selection of these values can lead to poor performance, such as slow response, oscillations, or even instability. Through careful tuning and implementation of the PID controller, the Ball and Beam system can be successfully stabilized, making it an excellent demonstration of the power and challenges involved in real-time control systems.

8 Acknowledgement

We would like to thank Professor Vineet Vashista for his valuable guidance and support throughout this project. His knowledge made a huge difference in our understanding and learning. We are also grateful to our Teaching Assistants, Rajdeep singh devra and Jenish Chauhan, for their help and dedication during our regular review meetings. Their support was key in helping us overcome many challenges.

References

[1] https://ctms.engin.umich.edu/CTMS/?example=BallBeam&section=SystemModeling

[2] https://pubs.sciepub.com/acis/3/1/1/

[3] https://www.tandfonline.com/doi/full/10.1080/21642583.2019.1575297